Amendments to the Specification

Please replace the paragraph that begins on Page 1, line 5 and carries over to Page 2, line 1 with the following marked-up replacement paragraph:

-- The present invention is related to U. S. Patent _____ (serial number ~~09/~~ 09/956,276), which is titled "Dynamic, Real-Time Integration of Software Resources through Services of a Content Framework", and U. S. Patent _____ (serial number ~~09/~~ 09/956,268), which is titled "Programmatic Management of Software Resources in a Content Framework Environment", both are which are commonly assigned to International Business Machines Corporation and which were filed concurrently herewith. --

Please replace the paragraph that begins on Page 21, line 15 and carries over to Page 22, line 8 with the following marked-up replacement paragraph:

-- In addition to specifying the public interface, the portlet's deployment port type information must be created (Block 610). An example of the deployment port type is illustrated in Fig. 4A. If the port type information is programmatically generated, for example using WSTK, this process comprises parsing the software resource's class definition to locate its methods providing the operations of the predefined deployment port type and to identify the parameters and parameter types of each method, and then generating <message> and <operation> elements, as well as <part> elements and <type> definitions as necessary. The composer may be asked to provide input during this process, as needed, such as populating the parameters for the messages of the operations. (For example, to populate the "displayIcon16x16Output" parameter of the "getDisplayIcon16x16Output" parameter message, the composer might specify the

Uniform Resource Locator, or "URL", of a particular image file.) The portlet's system port type information is also created (Block 620), in a similar manner, resulting in a WSDL document such as that shown in Fig. 4B. (In preferred embodiments, the deployment and system interface definitions are placed in separate WSDL documents as shown in the examples of Figs. 4A and 4B because they describe different behavior.) --

Please replace the paragraph on Page 23, lines 11 - 18 with the following marked-up replacement paragraph:

-- The web service composition tool preferably provides a portlet palette for use in this modeling operation, where registered portlets for a particular taxonomy or category are presented on the palette. The service composer then creates a new web service using the composition tool, for example by right-clicking on a web service icon to display its available methods and then using drag and drop operations to position ~~selection~~ selected method invocations as operations for carrying out a service. Fig. 9, discussed below, illustrates logic with which information may be gathered for use by the web service composition tool, including locating the appropriate web services to use when constructing the palette. --

Please replace the paragraph that begins on Page 29, line 1 and carries over to Page 30, line 4 with the following marked-up replacement paragraph:

-- Returning now to the discussion of Fig. 7, once the composer completes the model of service interaction as a directed graph, at Block 710 the composer declares a service provider port type for the newly-aggregated service. In order to define this public interface, the composer

may select operations to be exposed, or exported, within port types of the service of the composition, thereby identifying the public port type. The composer may select as many operations for exporting as necessary, and may also define new operations if desired. (For example, the composer might decide to define a new name for an operation, and provide a one-to-one mapping for that new name to the old name.) These exposed operations specify the means for invoking the new service. For example, with reference to the sample directed graph in Fig. 5, the composer would specify (at least) the operation "receivePurchaseOrder" corresponding to node 530 as a public interface. The composer may also provide a data mapping between an exposed operation and an internal service port type operation. For the sample directed graph in Fig. 5, having a public interface of "receivePurchaseOrder", the composer might specify (for example) a data mapping that converts data to a particular format required by this operation, such that an input parameter named "inputPurchaseOrder" adhering to a predefined type "purchaseOrder" would be available to the service upon invocation. Providing the data mapping may comprise identifying a canned transformation component, such as an integer-to-float transform; identifying a stylesheet which contains an appropriate transformation, such as an XSLT (Extensible Stylesheet Language Transformations) stylesheet; identifying customized transformation logic, which may for example convert one complex data type to another; and so forth. (Refer to the discussion of ~~Fig. 11~~ Figs. 11A - 11C, below, for more information about declaring the interface to a web service intermediary and potential web service software resources, and about providing transformation information.) The process of declaring the exposed port types (i.e. operations) results in the new service provider type for the aggregated composition. Thus, this new composition is itself a portlet proxy, or web service intermediary,

that can be used as a building block in further compositions. --

Please replace the paragraph on Page 30, lines 5 - 13 with the following marked-up replacement paragraph:

-- The composer also provides a mapping for the deployment port type and for the system port type, if applicable, such that the resulting web service will be able to be used within a web service composition tool and will be able to be programmatically managed by the portal platform as disclosed herein. This is preferably accomplished by prompting the composer (using a menu-driven approach, for example) to identify operations and data mappings that will then be represented in WSFL markup language <export> and <dataLink> elements, which enable providing a new interface and doing internal mapping for that interface. (Refer to section 4.5.3, "Data Links and Data Mapping", and section 4.6.4, "Exporting Operations", of the WSFL specification for more information on the syntax and semantics of the <export> and <dataLink> elements.) --

Please replace the paragraph that begins on Page 30, line 14 and carries over to Page 31, line 7 with the following marked-up replacement paragraph:

-- In some cases, it may be desirable to generate the deployment and/or system port types (and similarly, the public interface of the portlet proxy) without reference to a particular target software resource. For example, suppose that run-time quality of service is an important factor in choosing a service provider for some service such as credit card processing. To provide a more generic interface which will programmatically adapt to one of several target service providers

after a run-time service selection, according to the present invention, the composer may be asked to provide input for use in creating the WSDL document (see the description of Blocks 600 through ~~620; this~~ 620); this process may be assisted by parsing class definitions and having the composer point to the right operations. The examples in Figs. 11A and 11B show signatures that might exist for two candidate software resources for a credit card processing operation. The example in Fig. 11C shows a sample signature that might be designed as a superset which includes all parameters for both possible candidates. The manner in which a run-time mapping of input parameter values to the parameters required by the selected resource occurs is described below. --

Please replace the paragraph that begins on Page 44, line 18 and carries over to Page 45, line 3 with the following marked-up replacement paragraph:

-- While the preferred embodiments of the present invention have been described, additional variations and modifications in those embodiments may occur to those skilled in the art once they learn of the basic inventive concepts. Therefore, it is intended that the appended claims shall be construed to include [[both]] the preferred ~~embodiment~~ embodiments and all such variations and modifications as fall within the spirit and scope of the invention. --